KINEMATICS MODELLING OF ROBOT MANIPULATOR USING SOLIDWORKS

Compiled as one of the requirements of completing the Undergraduate Program at the Department of Electrical Engineering Faculty of Engineering

By:

MILZAM WAFI AZHAR

D400 154 007

PROGRAM STUDY ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2019

APPROVAL PAGE

KINEMATICS MODELLING OF ROBOT MANIPULATOR USING SOLIDWORKS

PUBLICATIONS JOURNAL

Arranged By:

MILZAM WAFI AZHAR

D400 154 007

This journal has been checked and approved by:

Supervisor

Dr. Muhammad Kusban S.T., M.T
NIK: 663

AFIRMATION PAGE

KINEMATICS MODELLING OF ROBOT MANIPULATOR USING SOLIDWORKS

By

MILZAM WAFI AZHAR

D400 154 007

Has been tested in front of The Council of Examiners
Faculty of Engineering
Muhammadiyah University of Surakarta
On July 15th 2019

The Council of Examiners:

1. **Dr. Muhammad Kusban S.T., M.T** (……………)
   (Head of Examiner)
2. **Dedy Ari Prasetya, S.T., M.Eng** (……………)
   (Member I of Examiner)
3. **Heru Supriyono, S.T., M.Sc ., Ph.D** (……………)
   (Member II of Examiner)

The Head of Faculty,

UNIVERSITAS MUHAMMADIYAH SURAKARTA FAKULTAS TEKNIK

Ir. Sri Sunarjono, M.T., Ph.D.
NIK. 682

TESTIMONY

I hereby declare that in this final project there is no work ever submitted to the degree of scholarship in a college and in my knowledge there is also no work or opinion that has been written or published Others, except in writing referenced in the manuscript and mentioned in the bibliography.

When there is a proven untruth in my statement above, I will be fully responsibility.

Surakarta, July 15th 2019

Author

MILZAM WAFI AZHAR

D400 154 007

KINEMATICS MODELLING OF ROBOT MANIPULATOR USING SOLIDWORKS

Abstrak

Manipulator robot memiliki 4 DOF (Degree of Freedom), semua sambungan bersifat revolute serta dilengkapi dengan pointer di end-effector. Proyek ini bertujuan untuk menganalisis kinematika maju dan terbalik dari robot manipulator yang didesain dengan menggunakan SolidWorks untuk mendapatkan model 3-D dan menggunakan *Sismscape Multibody Link* generasi pertama untuk mengekspor CAD agar mendapatkan format file XML dan STL. Kemudian mengeksekusi file XML dengan Jendela perintah MATLAB untuk memperoleh diagram blok robot di Simulink. Mengembangkan GUI (Graphical User Interfaces) menggunakan MATLAB GUIDE untuk mengendalikan lengan dan menentukan kinematika maju dan terbaliknya. Pertama, membuat skema robot yang akan digambar dengan semua dimensi yang berlabel. Selanjutnya, tetapkan bingkai koordinat ke diagram robot sesuai dengan algoritma DH (*Denavit-Hartenberg*) dan kinematika maju lengkap akan dianalisis pada robot menggunakan persamaan *Homogeneous Transformation Matrix* yang memerlukan notasi. Hasil akhir dari *Homogeneous Transformation Matrix* akan menghasilkan orientasi dan posisi dari *end-effector*. Selain itu, analisis kinematika terbalik juga akan dilakukan pada dua titik yang diasumsikan, misalnya, mengangkat dan menempatkan dalam ruang kerja dengan menggunakan simulasi GUI sehingga akan diperoleh nilai sudut dari setiap jointnya.

Kata Kunci: Algoritma dh (denavit-hartenberg), homogeneous transformation matrix, kinematika robot, matlab gui, simscape multibody link.

Abstract

The robot manipulator had 4 DOF (Degree of Freedom), all joints are revolute as well as equipped with pointer at the end-effector. This project aims to analyse the forward and inverse kinematics of the robot manipulator that has design with SolidWorks to get the 3D model and using *Simscape Multibody Link* first generation to export the CAD file to get the XML and STL file format, then execute the XML file with MATLAB command window to obtain the block diagram of the robot. Developing the GUI (Graphical User Interfaces) using MATLAB GUIDE to controlling and determining the robot kinematics value. Firstly, a schematic of the robot will be drawn with all the dimensions labelled. Next, assign coordinate frames to the robot diagram according to DH algorithm and complete forward kinematics will be analysed for the robot using *Homogeneous Transformation Matrix* that required notation. Final result of *Homogeneous Transformation Matrix* will produce the orientation and position of the end effector. Besides, an inverse kinematics analysis will also carry out for two assumed points, for example, pick and place within the workspace by using GUI simulation so that will obtain angles value of each joint.

Keywords: dh algorithm (denavit-hartenberg), homogeneous transformation matrix, robot kinematics, matlab gui, simscape multibody link.

1. INTRODUCTION

The development of science and technology, especially in the field of electronics, has developed rapidly. These developments are in line with the increasing demands of the community for high-quality goods produced by the industry. To support high quality, many industrial processes switch from manual systems to automatic systems, with the role of human being getting smaller, it requires a performance support tool in the industry. One of the tools is a robot that can replace human jobs related to repetitive activities.

Manufacturing processes such as arc welding, spray painting, assembly, pick and place, cutting, milling, drilling, are required the robot manufacturing. Most popular type of robot is the industrial robot. Type of robot manipulator are rectangular, cylindrical, spherical, revolute and horizontal jointed (Gouasmi, M., Ouali, M., Fernini, B., & Meghatria, M, 2012).

One type of robot used in the industrial world is the arm robot. With the increasingly important role of arm robots in the industrial world, many students are interested to learning arm robots. Meanwhile, to make arm robots in the industrial world scale is not cheap. For this reason, needed an arm robot simulator that can used to determine the direction and movement of the *end effector*. To determine the direction and movement of the arm robot, by evaluating and analysis the robot kinematics. Forward kinematics is the process of calculating the orientation and position of the *end effector* based on the angles at the joint. Whereas inverse kinematics is the opposite, given the position of *end effector*, what will be calculate is how many angles must be changed for each joint to be able to reach the *end effector* position.

So, in this Final Project, the author will design and create a 4 DOF (Degree of Freedom) robot arm simulator which is a representation of arm robots for industry that can later be used for the Education sector. This arm robot simulator is designed using the 3D SolidWorks design application and exported using the first generation *Simscape Multibody Link* to get the XML file format and open it in Simulink for MATLAB to obtain the block diagram from the arm robot. Robot arm simulation based on the GUI (Graphical User Interfaces) is made using GUIDE (Graphical User Interfaces Developer Environment) in MATLAB to control and determine the *end effector* position and the angle of each joint.

2. METHOD

This study will make a arm robotic simulation program to analyze and apply kinematic equations with the following stages of research:

1) System planning

2) Modelling by SolidWorks.

3) Robot kinematics.

4) *Simscape Multibody Link* and dynamic study.

5) Simulation methodology.

6) Simulation result and discussion.

The stages of the research are shown in figure 1.

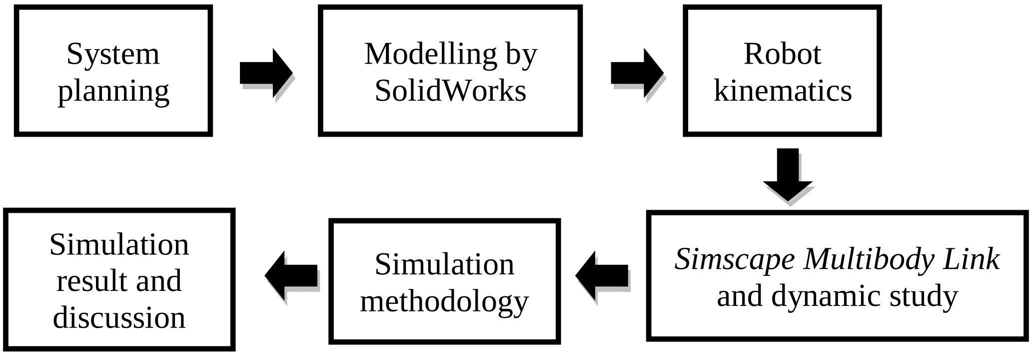


Figure 1. Stages of Research

2.1 System Planning

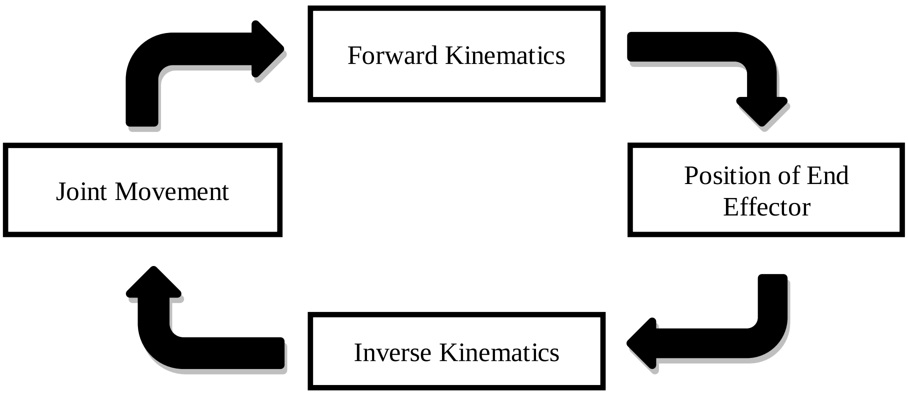
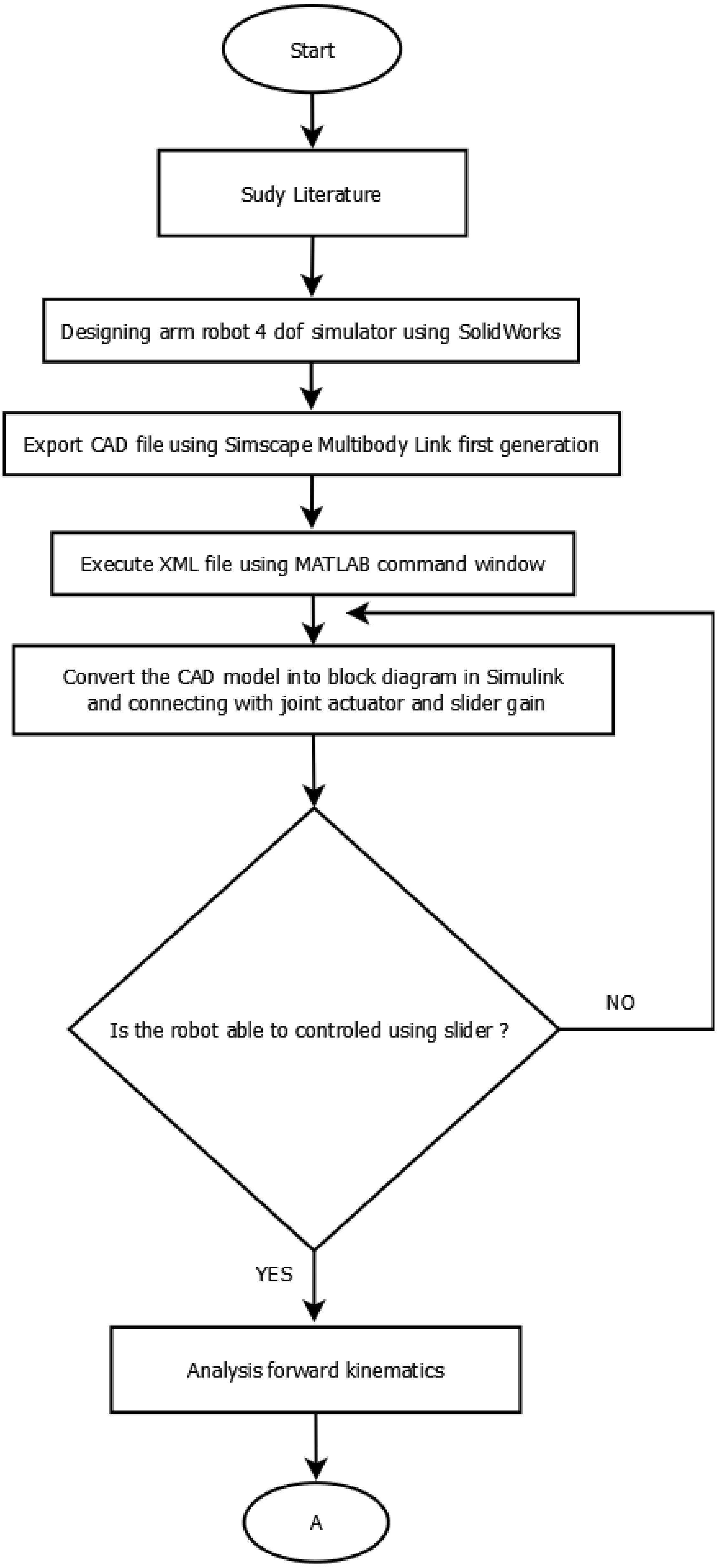


Figure 2. Block Diagram system

The block diagram in Figure 2 explains how the kinematics system of arm robot evaluated. There are 2 variable of robot kinematics forward and inverse kinematics.


Start
Sudy Literature
Designing arm robot 4 dof simulator using SolidWorks
Export CAD file using Simscape Multibody Link first generation
Execute XML file using MATLAB command window
Convert the CAD model into block diagram in Simulink and connecting with joint actuator and slider gain
Is the robot able to controled using slider ?
NO
YES
Analysis forward kinematics
A

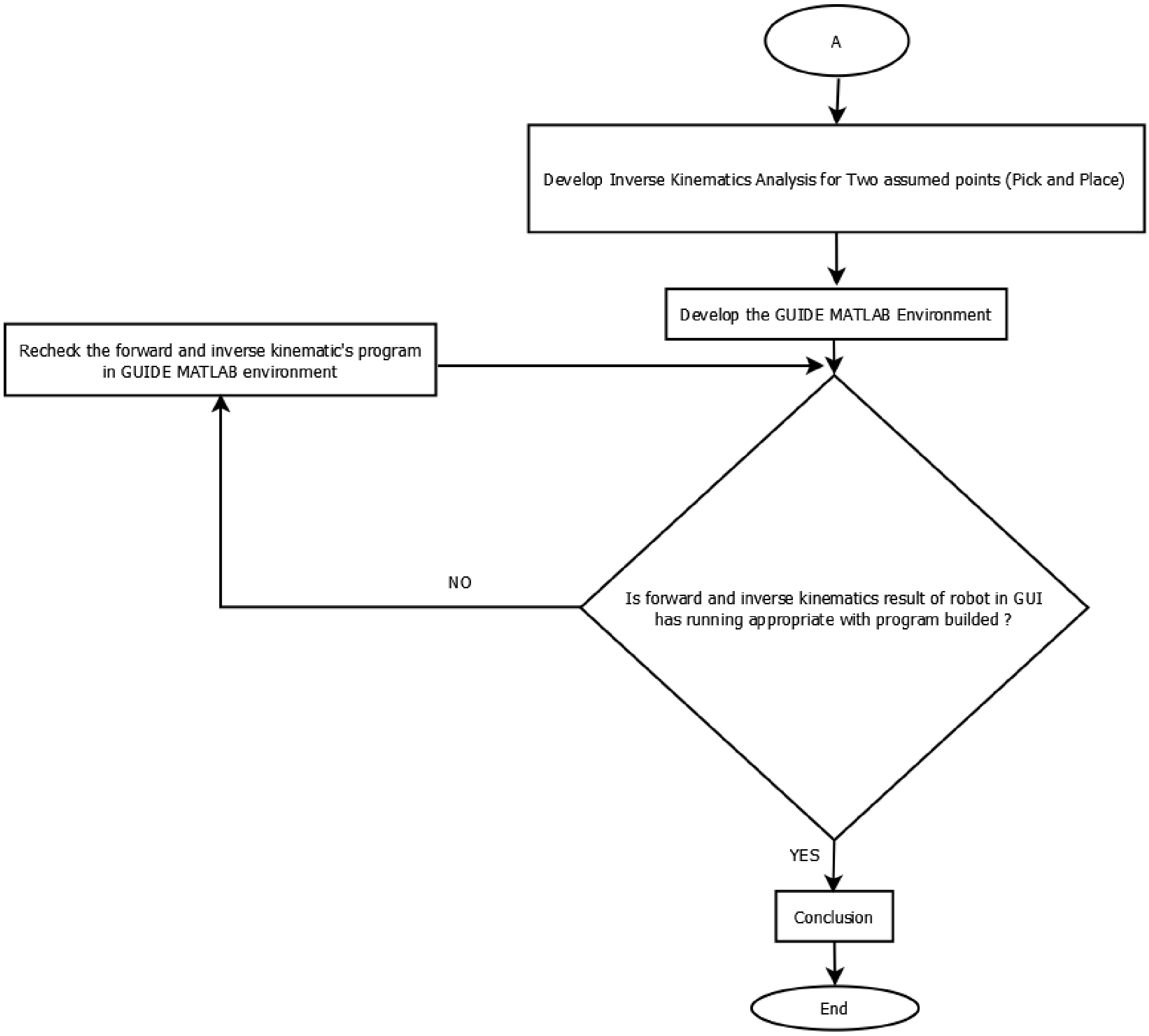


Figure 3. Flowchart system

Flowchart of the arm robot simulation system is explained in figure 3, the first step is designing 3D model of the robot using SolidWorks. Then export the model to simulink/MATLAB using simscape multibody link first generation in the form of XML and STL format file. In command window of MATLAB execute the file in order to get the block diagram in simulink. Modify the block diagram by adding several sensors, actuator, slider gain and connector to controlling the arm robot. Evaluate the kinematics robot and verification using robotic toolbox for MATLAB to make sure that the kinematics program is correct. Build the GUI (Graphic User Interface) using GUIDE MATLAB Environment to make easier monitoring and controlling the arm robot.

2.2 Modelling by SolidWorks

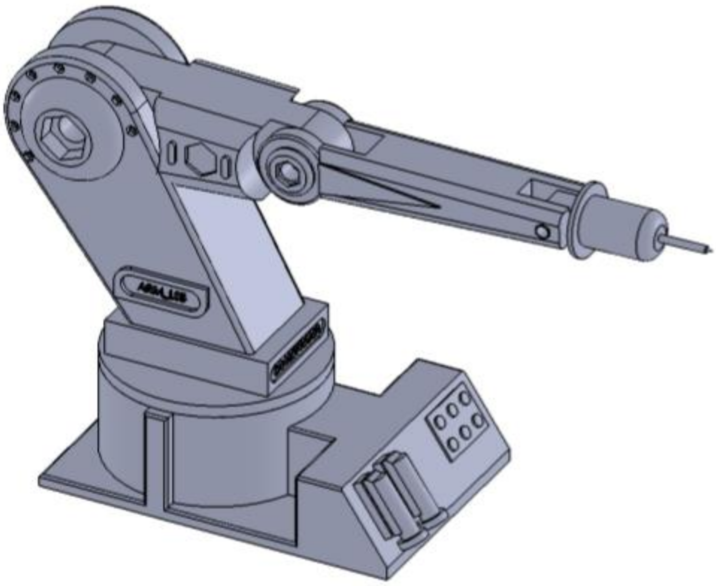


Figure 4. 3-D design model of robot in SolidWorks

SolidWorks is what we call as "parametric" solid modelling using for 3-D design modelling. Parametric means that the dimension have relationship with one another and able to changed when design process and automatically changes the solid part and related documentation (Mariappan, S. M., & Veerabathiran, A, 2017).

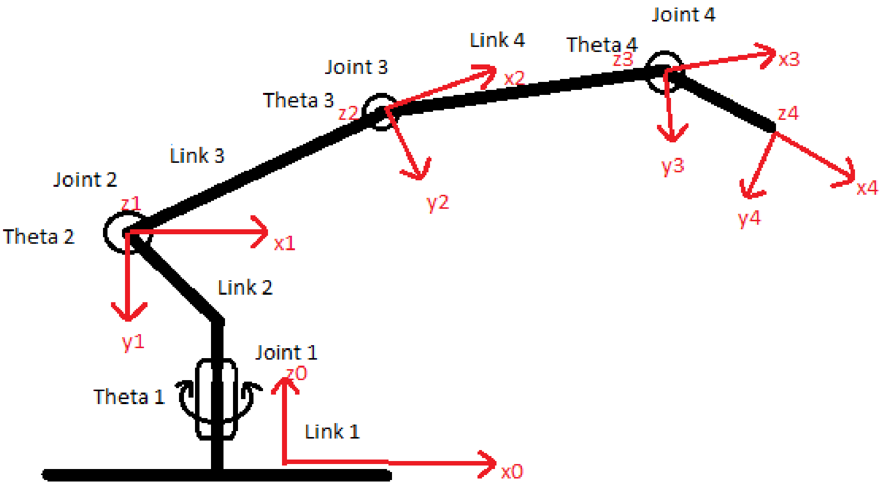


Figure 5. Coordinate Frame of 4 DOF Arm Robot

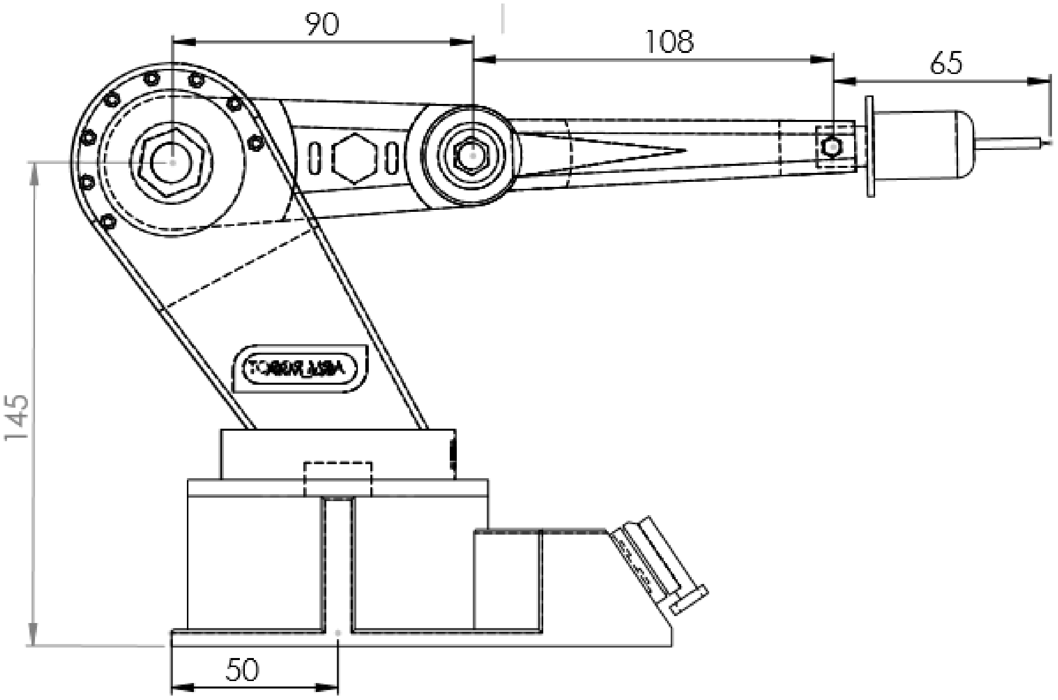


Figure 6. Schematic of 4 DOF Arm Robot

Figure 6. Shows the schematic of arm robot 4 DOF with all dimensions labeled. After all dimensions is determined, then input the value of each notation appropriate with schematics labeled in DH parameters Table 1.

Each notation has provisions as follows:

a. i is link to i
b. i-1 is link to i-1
c. α_1 is angle between z_1 and z_{i-1} in axis x_i
d. a_i is distance between x_{i-1} and x_i in axis z_{i-1} measured along axis x_i
e. θ_i is angle between x_{i-1} and x_i in axis z_i
f. d_i is distance between x_{i-1} towards x_i measured along axis z_i.

2.2.1 Forward Kinematics

Denavit Hartenberg Parameters

$L_1 = 50$; $L_2 = 90$; $L_3 = 108$; $L_4 = 65$; $d_1 = 145$

Table 1. DH Parameters

Joint	Θ_n	α_n	a_n	d_n
1	Θ_1	-90	L_1	d_1
2	Θ_2	0	L_2	0
3	Θ_3	0	L_3	0
4	Θ_4	0	L_4	0

Homogeneous Transformation Matrix Equation

$$^{i-1}A_i = \mathit{Trans}(0, 0, d_i)\mathit{Rot}(z, \theta_i) \quad \mathit{Trans}(a_i, 0, 0) \quad \mathit{Rot}(x, a_i) \tag{1}$$

$$^{1-i}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & \alpha_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T1 = \begin{bmatrix} cos\theta_1 & 0 & -\sin\theta_i & 50\cos\theta_1 \\ \sin\theta_1 & 0 & \cos\theta_1 & 50\sin\theta_1 \\ 0 & -1 & 0 & 145 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$T2 = \begin{bmatrix} cos\theta_2 & -\sin\theta_2 & 0 & 90\cos\theta_2 \\ \sin\theta_2 & cos\theta_2 & 0 & 90\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$T3 = \begin{bmatrix} cos\theta_3 & -\sin\theta_3 & 0 & 180\cos\theta_3 \\ \sin\theta_3 & cos\theta_3 & 0 & 180\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$T4 = \begin{bmatrix} cos\theta_4 & -\sin\theta_4 & 0 & 65\cos\theta_4 \\ \sin\theta_4 & cos\theta_4 & 0 & 65\sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$T = T1*T2*T3*T4 \tag{7}$$

By multiplying the equation (3, 4, 5, 6) we obtained the forward kinematics equation (8).

Forward Kinematics Equation

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & Px \\ r_{21} & r_{22} & r_{23} & Py \\ r_{31} & r_{32} & r_{33} & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$px = (\cos(\theta_1) * 130*\cos(\theta_2+\theta_3+\theta_4) + 216*\cos(\theta_2+\theta_3) + 180*\cos(\theta_2) + 100))/2 \quad (9)$$

$$py = (\sin(\theta_1) * 130*\cos(\theta_2+\theta_3+\theta_4) + 216*\cos(\theta_2+\theta_3) + 180*\cos(\theta_2) + 100))/2 \quad (10)$$

$$pz = 145 - 108*\sin(\theta_2+\theta_3)-90*\sin(\theta_2) - (130*\sin(\theta_2+\theta_3+\theta_4))/2 \quad (11)$$

The result from equation (8) obtained the position of *end effector* of the robot shows in equations (9, 10 & 11) in cartesian coordinate (Px, Py & Pz).

2.2.2 Inverse Kinematics for The End-Effector

Inverse kinematics of the robot can be obtained by using many methods, one of them is using algebraic approach. This method according to the result of forward kinematics (Px, Py & Pz).

$$\theta_1 = \text{atan2}\ (py,px) \quad (12)$$

$$\theta_2 = atan2\ (A,B) \quad (13)$$

$$\theta_3 = \text{atan2}\ (\sin(\theta_3), \cos(\theta_3)) \quad (14)$$

$$\theta_4 = -\theta_2 - \theta_3 \quad (15)$$

The equation (12, 13, 14 & 15) is final calculation result of inverse kinematics of the robot. There are 4 joint angles in robot as notation ($\theta_1, \theta_{2,} \theta_3, \theta_4$). After all joints angle are obtained, write the equation in GUI program of inverse kinematics part.

2.3 Simulation Methodology

The 3D model of the robot was exported from SolidWorks environment to MATLAB/ Simscape Multibody Link environment in the form of XML and STL file through Simscape Multibody Link first generation. The XML file of the model executed using the MATLAB command window. The CAD model of the robot will converted into a block diagram in Simulink shows in Figure 7. By adding some part such as joint actuator and slider thee robot would be able to controlled.

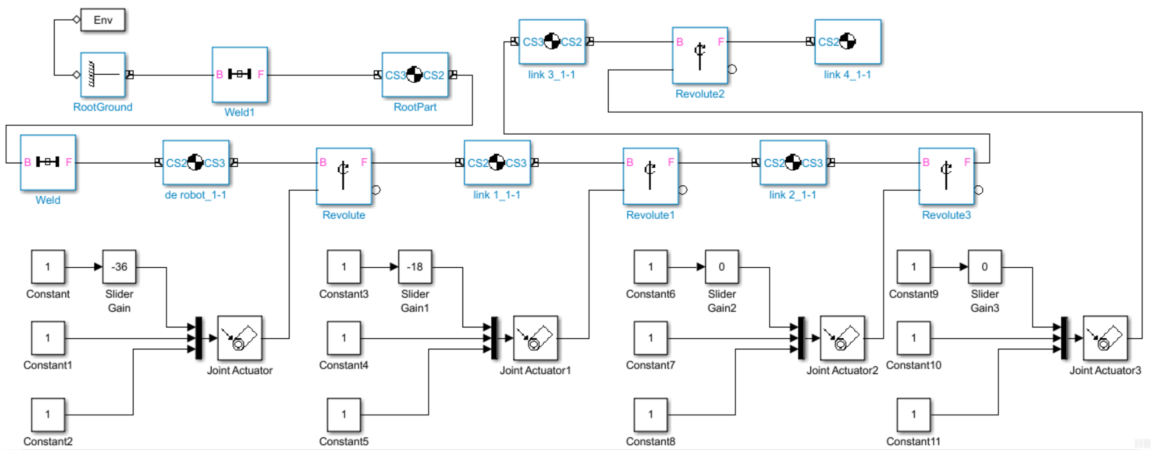


Figure 7. Simscape Multibody Link model

3. RESULT AND DISCUSSION

By developing the simulation system based on MATLAB, the forward kinematics and inverse kinematics of arm robot could be evaluated. The GUI is developed by using MATLAB GUIDE Environment. The forward and inverse kinematics interface is shown in figure 8.

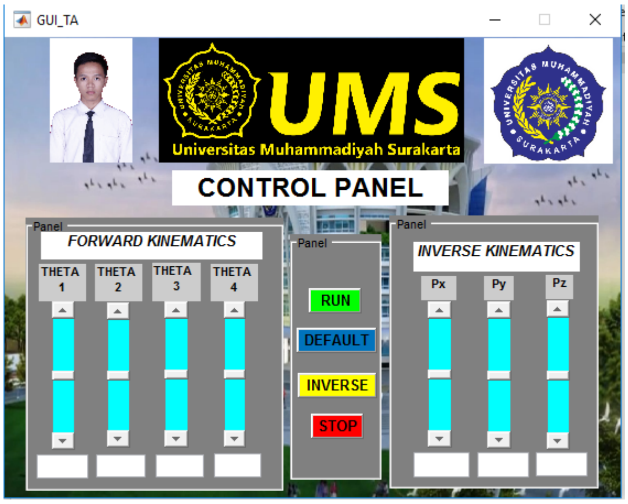


Figure 8. GUI forward and inverse kinematics

3.1 Default Position Result

The position and orientation (Px, Py, Pz) of the robot is editable. Default position of the robot is on zero position, and the end-effector of the robot in three dimensions space is (313, 0, 145) shown in figure 9.

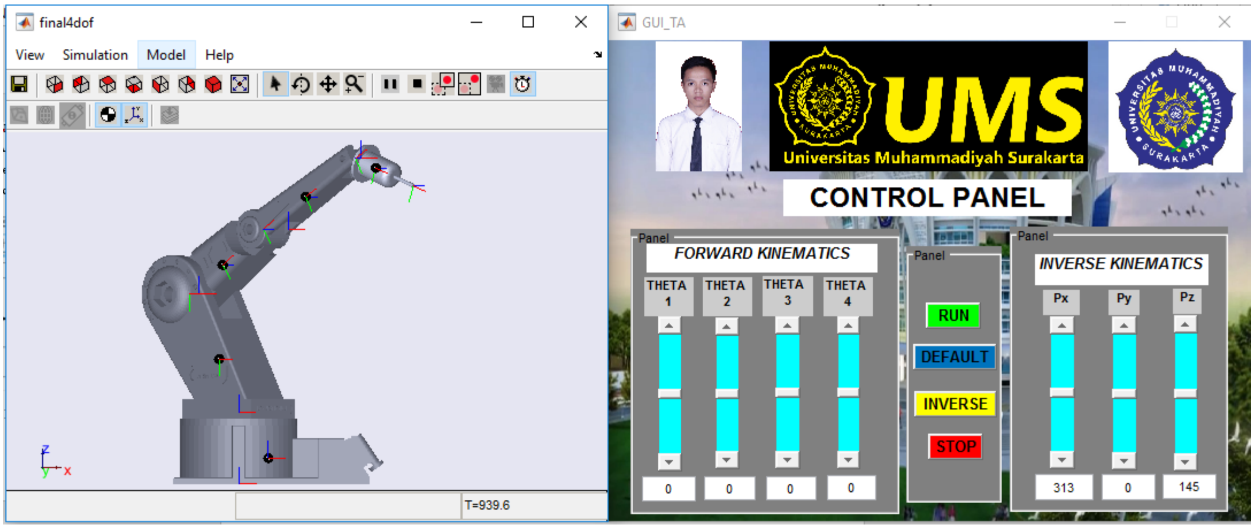


Figure 9. Robot arm interface at default position

To starting the simulation, user just run the program in MATLAB then the GUI will display. Click *run* button on GUI automatically will running and open the block diagram in Simulink. After the program is compiled, click on *default* button and the robot will get the zero position. The users able to change the value of angels (θ_1, θ_2, θ_3, θ_4) to evaluates the forward kinematics of robot by changing the sliders, then the position of end-effector (Px, Py, Pz) will obtained and the robot position will change as well. To calculate the inverse kinematics of the robot (θ_1, θ_2, θ_3, θ_4), change the value of (Px, Py, Pz) by move the slider then click the *inverse* button. The *stop* button will stop the program and close the GUI interface.

3.2 Forward Kinematics Result

The forward kinematics interface that shows in figure 10, the input angles value is (-72, 14.4, -72, 100.8) in degrees, through forward kinematics the position of the end-effectors is (-122.1962, -32.0664, 197.6632).

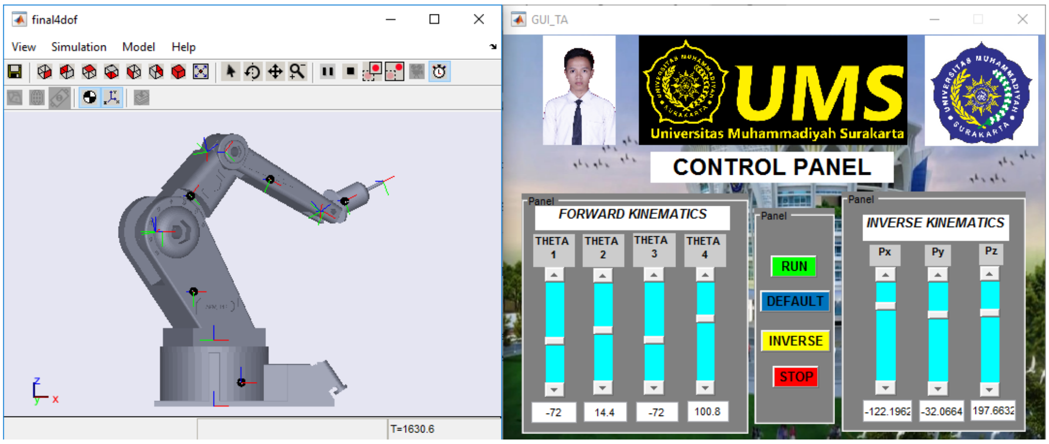


Figure 10. Forward kinematics of arm robot

3.3 Inverse Kinematics Result

The inverse kinematics interface is shown in Figure 11 & 12, the inverse calibration method can also be applied to get arm value based on the results of the forward kinematics process and directly the pick and place inverse kinematics of point (A to B) can be defined.

1) Point A (Pick)

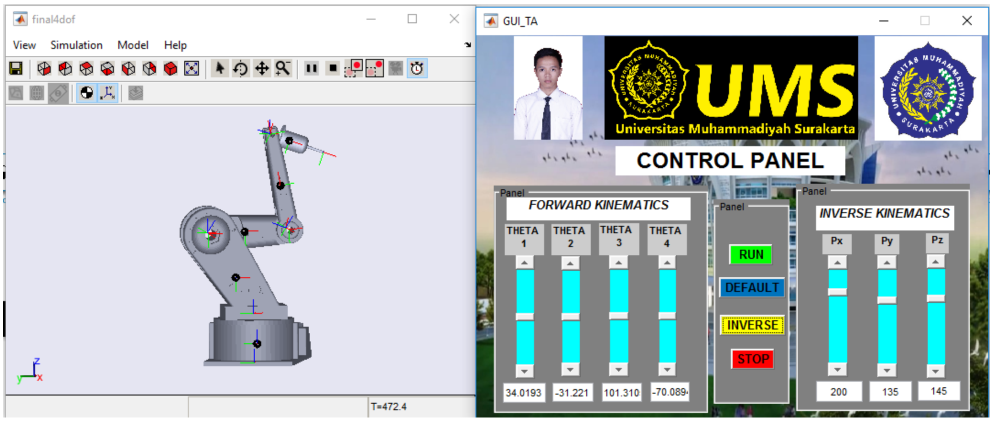


Figure 11. Point A of inverse kinematics result

2) Point B (Place)

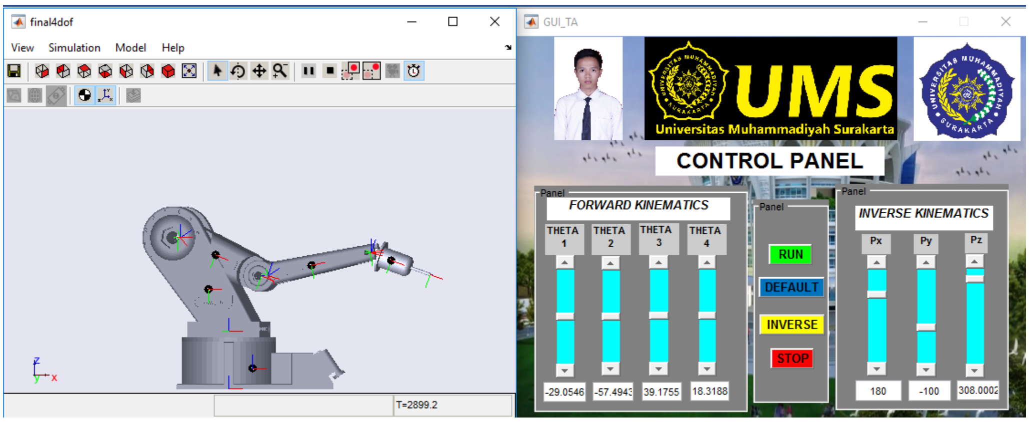


Figure 12. Point B of inverse kinematics result

The position of the end-effectors at point A is (200, 135, 145) and point B is (180, -100, 308). The four joint angles at point A are (34.0193, -31.221, 101.3105, -70.0894) and point B is (-29.0546, 57.4943, 39.1755, 18.3188) in degrees through the inverse kinematics equations. If the pose of the end-effectors cannot reach, the program will enable to run and the robot is not move.

Table 2. Position and Orientation Result of The Robot

Condition		Θ_1 (Degree)	Θ_2 (Degree)	Θ_3 (Degree)	Θ_4 (Degree)	Px	Py	Pz
Default Position		0	0	0	0	313	0	145
Forward Kinematics		-72	14.4	-72	100.8	122.19	-32.06	197.66
Inverse Kinematics	Pick	34.01	-31.2	101.31	-70.08	200	135	145
	Place	-29.05	-57.49	39.17	18.31	180	-100	308

Table 2 Shows the summary of robot kinematics. there are four condition of position and orientation of the robot. Default position shows in Figure 9, where the robot has initial position. Forward kinematics shows in Figure 10 And inverse kinematics shows in Figure 11 and Figure 12 by assuming two points as pick and place.

4. CLOSING

The purpose of this project is for educational tool which can be used for robotics study, the student will easier to analysis the kinematics model of arm robot by using this simulator. Improving the implementation of kinematics solution practically. Evaluating the inverse kinematics is more complex than forward kinematics when there is no single analytical solution. There are several methods to evaluate the inverse kinematics one of them on this research by using algebraic approach method. The reason using this method because algebraic approach is more efficient.

ACKNOWLEDGMENT

The author thanks a lot to people who gave motivation and help in arranging final assignment as follows:

1) Parents who always pray, encourage and support in doing this final assignment.
2) Dr. M. Kusban as the supervisor who gives his guidance so that this final project can have maximum result.

3) Head of Department of Electrical Engineering, Muhammadiyah University of Surakarta, Mr. Umar, S.T, M.T and all electrical lecturers at the Department of Electrical Engineering Muhammadiyah University of Surakarta.

4) To international classmate who always provide input and direction when working on the final assignment.

REFERENCES

Corke, P. (2007). “A simple and systematic approach to assigning denavit hartenberg parameters,” IEEE Transactions on Robotics, vol.23, pp.590-594

Gouasmi, M., Ouali, M., Fernini, B., & Meghatria, M. (2012). "*Kinematic modelling and simulation of a 2-R robot using solidworks and verification by matlab/simulink"*. International Journal of Advanced Robotic Systems, *9*, 1–13. https://doi.org/10.5772/50203

Mariappan, S. M., & Veerabathiran, A. (2017). "*Modelling and simulation of multi spindle drilling redundant SCARA robot using SolidWorks and MATLAB/SimMechanics"*. Revista Facultad de Ingeniería Universidad de Antioquia, (81), 63–63. https://doi.org/10.17533/udea.redin.n81a06

Serrezuela, R. R., Chavarro, A. F. C., Cardoso, M. A. T., Toquica, A. L., & Martinez, L. F. O. (2017)."*Kinematic modelling of a robotic arm manipulator using MATLAB"*. ARPN Journal of Engineering and Applied Sciences, *12*(7), 2037–2045.

X. Xiao, Y. Li, and H. Tang, (2013). "*Kinematics and interactive simulation system modeling for robot manipulators"*, Proceeding of the IEEE International Conference on Information and Automation, Yinchuan, China, pp.1174-1182.

Zhao, H., Lu, Z., Liu, C., & Wang, H. (2017). "*Model and simulation of the Mitsubishi RV-M1 robot using MATLAB"*. 2016 IEEE International Conference on Signal and Image Processing, ICSIP 2016, 582–585. https://doi.org/10.1109/SIPROCESS.2016.7888329